Computers and Mathematics with Applications 57 (2009) 1985–1994

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Bi-level programming model and hybrid genetic algorithm for flow interception problem with customer choice[☆]

Jun Yang[a,*], Min Zhang[b], Bo He[a], Chao Yang[a]

[a] School of Management, Huazhong University of Science and Technology, Wuhan, 430074, China

[b] School of Information Management, Wuhan University, Wuhan, 430072, China

ARTICLE INFO

Keywords:
Hybrid intelligent algorithm
Location
Path choice
Simulation

ABSTRACT

This paper investigates how to optimize the facility location strategy such as to maximize the intercepted customer flow, while accounting for "flow-by" customers' path choice behaviors and their travel cost limitation. A bi-level programming static model is constructed for this problem. An heuristic based on a greedy search is designed to solve it. Consequently, we proposed a chance constrained bi-level model with stochastic flow and fuzzy trip cost threshold level. For solving this uncertain model more efficiently, we integrate the simplex method, genetic algorithm, stochastic simulation and fuzzy simulation to design a hybrid intelligent algorithm. Some examples are generated randomly to illustrate the performance and the effectiveness of the proposed algorithms.

1. Introduction

Traditional location–allocation models, such as the maximal covering location model (MCLM) and the p-median model, aim to locate network facilities to optimally serve demand expressed as weights at nodes [1–4]. Nowadays, many customers purchase services as part of routine pre-planned trips, i.e., the daily commute to and from home and the workplace, instead of making a special-purpose trip to obtain a service. Such facilities include convenience stores, gas stations, ATM machines, drugstores, laundries and restaurants. Thus, as the purchasing behavior changes, there are cases where demand in a network is now expressed as flows, rather than nodes.

To solve these types of facility locations in a network where demand is not expressed at nodes, but is exerted by traffic flowing between origins and destinations, Hodgson [5] and Berman [6] presents the flow interception problem (FIP) and developed a heuristic greedy algorithm to solve the FIP. The basic problem of FIP [5,6] is to locate m facilities to intercept as much flow as possible from a given set of pre-existing flows on the network. It assumes the "interception" occurs if a flow passes through at least one facility. The focus is on maximizing the total consumption of the service by "flow-by" customers traveling on preplanned paths (e.g. daily commute). Based on the basic FIP,they also published a series of studies for a class of FIP [7–9].

During the last two decades, uncertainty theory has experienced spectacular growth and is a hotspot in location science. The present papers recognize uncertainty in the demand or population at the nodes of the network or the different travel time between the nodes, which depend on the time of the day or day of the week. Uncertainty theory has been considered in the traditional location models (P-median, center problem, set-covering problem) [10–13]. In these FIP modes above, "flow-by" customers which are static only travel on preplanned paths. Nowadays some probabilistic models of locating

☆ The project was supported by the National Natural Science Foundation of China (Grant No. 70601011) and the Program for New Century Excellent Talents in University of China under the Grant No. NCET-06-0653 and the Postdoctoral Foundation of HUST.

* Corresponding author.
E-mail addresses: jun_yang@mail.hust.edu.cn (J. Yang), ibananai@163.com (M. Zhang).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.035

flow-capturing facilities are investigated [14–16]. In the probabilistic models, pre-planned paths are not known and only information on the fractions of customers traveling from any node to any adjacent node(transition probabilities) and the initial distribution of customers among nodes is available.Then the theory of Markov decision processes is applied for the analysis [14].

According to the relationship between customers and facilities, J. Yang categorizes the flow interception problem into three types: cooperative FIP, independent FIP and opposite FIP [17]. In independent FIP, for facility managers, the objective function is to maximize the intercepted "flow-by" customers; but from the "flow-by" customers perspective, they are concerned with two factors for choosing the paths from their origin to their destination. On one hand, they desire to obtain services from facilities on their trip. On the other hand, the expected trip cost (travel time) cannot be above the threshold level that they can bear. This problem can be described within a game theoretic framework as leader–follower or Stackelberg game [18]. Thus, a bi-level model for this problem is formulated in this paper.

Due to the NP-hardness of bi-level programming problem [19], a number of authors proposed various exact algorithms for solving it [20–22]. As for researches on computational methods using meta-heuristics for bi-level programming problem, Liu designed a genetic algorithm for solving a Stackelber–Nash equilibrium of nonlinear multilevel programming with multiple followers in which there might be information exchange among the followers [23]. Gendreau, Marcotte and Savard proposed an adaptive search method related to the Tabu Search meta-heuristic to solve the linear bi-level programming problem [24]. Li, Tian and Min developed a new algorithm framework based on particle swarm optimization for solving general bi-level programming problem, which combines two variants of PSO to solve the upper-level and lower-level programming problems interactively and cooperatively [25]. Takeshi and Hideki formulated defensive location problem as bi-level zero-one programming problems and proposed an algorithm based upon tabu search methods [26].

In this paper, we investigate how to optimize the facility location strategy such as to maximize the intercepted customer flow, while accounting for "flow-by" customers' path choice behaviors and their travel cost limitation. The purpose of this paper is to develop a bi-level model for this problem and to design meta-heuristic algorithms to solve it.

The rest of the paper is organized as follows. In Section 2, the problem and symbols used are introduced. And we construct a bi-level programming static model for this problem. A heuristic based on a greedy search is designed to solve this model in Section 3. In Section 4, we suppose customers of OD pairs be stochastic variables. And the customers in general choose their paths in order to obtain service as conveniently as possible, while satisfying the trip cost threshold level which is a fuzzy variable. Thus,on the basis of credibility measure, a bi-level chance constrained model for FIP is developed. For solving this model more efficiently, we integrate the simplex method, genetic algorithm, stochastic simulation and fuzzy simulation to design a powerful hybrid intelligent algorithm in Section 5. Finally, Section 6 provides some numerical examples generated randomly to illustrate the performance and the effectiveness of the proposed algorithm.

2. A bi-level programming model for the static FIP

2.1. The basic idea of bi-level programming for FIP

This FIP can be represented as a leader–follower game where the facilities location planner are leader, and the "flow-by" customers can freely chose their paths are the followers. It is assumed that the facilities location planning managers can influence, but cannot control the customers' path-choosing behavior. The customers make their decision in a customer optimal manner. This interaction game can be described as the following bi-level programming problem.

(U0) :

$$\max_x \; F(x, q) \tag{2.1}$$

subject to

$$G(x, q) \le 0 \tag{2.2}$$

where $q = q(x)$ is implicitly defined by

(L0)

$$\max_q \; f(x, q) \tag{2.3}$$

subject to

$$g(x, q) \le 0. \tag{2.4}$$

Obviously, the bi-level programming model consists of two sub-models, (U0) which is defined as an upper-level problem and (L0) which is a lower-level problem. F and x are the objective function and decision vectors of upper-level decision-makers or facility location planner, G is the constraint set of the upper-level decision vectors. f and q are the objective function and decision vectors of lower-level decision-makers or customers, g is the constraint set of the lower-level decision vectors.

The upper-level describes facility location problem and the lower-level model represents customers' path-choosing behavioral problem. In the bi-level Programming for FIP, the upper-level problem is to determine an optimal strategy for locating facilities with number limitation to capture the maximal "flow-by" customers. The lower-level problem represents customers assignment problem that describes customers' path-choosing behavior. Its objective function is to maximize the possibility of obtaining a service for each OD pair's customer flow.

2.2. The lower-level customer path-choosing behavior

It is worth emphasizing that the FIP problem must be solved with the network flow pattern. In general, the facility location strategy will definitely induce changes in customer flow over the network. Traditionally, the basic FIP models hypothesize that the flow on each path is given and fixed. In this bi-level programming model, the flow of each OD pair i ($i = 1, 2, \ldots, r$) is assumed to follow (L1) model under the given facility location project.

(L1)

$$\max_{q_{ij}} \sum_{j=1}^{p_i} u_{ij} q_{ij} \tag{2.5}$$

$$\sum_{j=1}^{p_i} c_{ij} q_{ij} \le c_i \tag{2.6}$$

$$\sum_{j=1}^{p_i} q_{ij} = 1 \tag{2.7}$$

$$0 \le q_{ij} \le 1 \tag{2.8}$$

where the parameters are:

$V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes on the network G;

$R = \{(v_{si}, v_{ti}),\ (i = 1, 2, \ldots, r)\}$ is the set of the given OD pair;

$Q_i = \{P_i^1, P_i^2, \ldots, P_i^{p_i}\}$ is the set of all paths linking node v_{si} and node v_{ti};

$u_{ij} = 1$, if at least one facility located on the path P_i^j; otherwise, $u_{ij} = 0$;

c_{ij} is the trip cost of path P_i^j ($j = 1, 2, \ldots, p_i$);

c_i means the expected trip cost threshold level of OD pair (v_{si}, v_{ti}).

The decision variables are defined as follows:

q_{ij} is defined as the probability of choosing the path P_i^j ($j = 1, 2, \ldots, p_i$) for the flow of the *OD* pair (v_{si}, v_{ti}) ($i = 1, 2, \ldots, r$).

The constraint set basically that: constraint (2.6) states trip cost of each OD pair is less than the trip cost threshold level. Constraint (2.7) and (2.8) are definitional, nonnegativity and conservation of the probability of choosing the path P_i^j($j = 1, 2, \ldots, p_i$) for the flow of the *OD* pair (v_{si}, v_{ti}) ($i = 1, 2, \ldots, r$).

2.3. The upper-level optimization problem

The facility planner at the upper-level is assumed to make the decisions about facility location and investment in order to maximize the total flow served by m facilities. The upper-level for the facility location problem ($U1$) can be expressed as follows:

($U1$)

$$\max_{x_k} \sum_{i=1}^{r} \sum_{j=1}^{p_i} f_i q_{ij} u_{ij} \tag{2.9}$$

$$\sum_{v_k \in P_i^j} x_k \ge u_{ij} \quad \forall (v_{si}, v_{ti}) \in R, \forall P_i^j \in Q_i \tag{2.10}$$

$$\sum_{k=1}^{n} x_k \le m \tag{2.11}$$

$$x_k \in \{0, 1\},\ u_{ij} \in \{0, 1\} \quad \forall v_k \in V, \forall (v_{si}, v_{ti}) \in R, \forall P_i^j \in Q_i \tag{2.12}$$

where the parameters are:

f_i is customer flow of OD pair (v_{si}, v_{ti});

m is the number of facilities to locate.

The decision variables are defined as follows:

$x_k = 1$, if a facility is located at node v_k; 0, otherwise;

$u_{ij} = 1$, if at least one facility is located on the path P_i^j; $u_{ij} = 0$, otherwise.

2.4. The heuristic based on greedy search

As we all known, a bi-level programming model is an NP-hard problem. To solve this model, we design the heuristic based on greedy search.

Step 1: Generate a feasible solution X for upper model (U1).

Step 2: Set $f_u = inf$ and $f_l = 0$.

Step 3: If $|f_u - f_l| < \epsilon$, stop, or else go to step 4.

Step 4: Using the simplex method to solve model (L1) with X, we can obtain the flow choosing path possibilities q_{ij}. That is to say, we get flow f_{ij} on every path of each OD pair. Let $f_l = \sum_{i=1}^{r}\sum_{j=1}^{p_i} f_i q_{ij} u_{ij}$, where u_{ij} is decided by X.

Step 5: When the flow on every path is known, this problem turns into a basic FIP problem. So a new solution X' can be obtained based on greedy search algorithm proposed by Berman [6]. Let $f_u = \sum_{i=1}^{r}\sum_{j=1}^{p_i} f_i q_{ij} u_{ij}$, where u_{ij} is decided by X'.

Step 6: Let $X = X'$, then go to step 3.

3. A chance constrained bi-level programming model for stochastic FIP with fuzzy trip cost threshold

3.1. Chance constrained bi-level programming model

In fact, the trip cost which every customer of each OD pair can bear or be satisfied with depends on conditions and circumstances. And the customer flow of each OD pair is not a certain number. Thus we assume the trip cost threshold level of each OD pair's flow be a fuzzy variable and flow of each OD pair be a stochastic variable which obey some stochastic distributions. With this assumption, we can formulate the lower-lever model (L2) with fuzzy trip cost threshold and the upper-lever model (U2) with stochastic customer flow.

(L2)

$$\max_{q_{ij}} \sum_{j=1}^{p_i} u_{ij} q_{ij} \tag{3.1}$$

$$\mathrm{cr}\left\{\sum_{j=1}^{p_i} c_{ij} q_{ij} \le \tilde{c}_i\right\} \ge \beta_i \tag{3.2}$$

$$\sum_{j=1}^{p_i} q_{ij} = 1 \tag{3.3}$$

$$0 \le q_{ij} \le 1 \tag{3.4}$$

where the parameters are:

β_i is desired credibility of satisfying the trip cost threshold level;

$\tilde{c}_i$, which is a fuzzy variable, means the trip cost threshold level of OD pair (v_{si}, v_{ti}).

Constraint (3.2) states that credibility of satisfying the trip cost threshold level of OD pair (v_{si}, v_{ti}) is not less than β_i.

(U2)

$$\max_{x_k} \bar{f} \tag{3.5}$$

$$\Pr\left\{\sum_{i=1}^{r}\sum_{j=1}^{p_i} f_i q_{ij} u_{ij} \ge \bar{f}\right\} \ge \alpha \tag{3.6}$$

$$\sum_{v_k \in P_i^j} x_k \ge u_{ij} \quad \forall (v_{si}, v_{ti}) \in R,\ \forall P_i^j \in Q_i \tag{3.7}$$

$$\sum_{k=1}^{n} x_k \le m \tag{3.8}$$

$$x_k \in \{0, 1\},\ u_{ij} \in \{0, 1\} \quad \forall v_k \in V,\ \forall (v_{si}, v_{ti}) \in R,\ \forall P_i^j \in Q_i \tag{3.9}$$

α predetermined by decision makers is confidence level at which it is desired the total served flow is not less than $\bar{f}$.

In this model (U2), the aim is to seek the optimal α-optimistic of this problem. Constraint (3.6) states that solution is feasible if and only if the probability measure of the event $\sum_{i=1}^{r}\sum_{j=1}^{p_i} f_i q_{ij} u_{ij} \ge \bar{f}$ is at least α.

3.2. Computing uncertain functions

By uncertain function we mean the functions with stochastic parameters and the functions with fuzzy variables. The function with stochastic parameters is

$$UF(1) : \Pr\left\{\sum_{i=1}^{r}\sum_{j=1}^{p_i} f_i q_{ij} u_{ij} \geq \bar{f}\right\} \geq \alpha. \tag{3.10}$$

In order to compute $\bar{f}$, we design a stochastic simulation as follows:

Step 1. Generate $\omega_1, \omega_2, \ldots, \omega_S$ from Ω according to the probability measure Pr, where S is a sufficiently large number;

Step 2. For each ω_s, we can obtain the value $F_s = \sum_{i=1}^{r}\sum_{j=1}^{p_i} f_i q_{ij} u_{ij}$, $s = 1, 2, \ldots, S$, respectively;

Step 3. Set S^* as the integer part of αS;

Step 4. Return the S^*th larger element in $\{F_1, F_2, \ldots, F_S\}$.

The function with fuzzy parameters is

$$UF(2) : \mathrm{cr}\left\{\sum_{j=1}^{p_i} c_{ij} q_{ij} \leq \tilde{c}_i\right\} \geq \beta_i. \tag{3.11}$$

If the fuzzy number $\tilde{c}_i$ is not continuous, we may use the following fuzzy simulation algorithm to get the approximately crisp equivalent of $UF(2)$:

Step 1. Randomly generate two numbers λ_1 and λ_2 such that $\mathrm{cr}\{\lambda_1 \leq \tilde{c}_i\} \geq \beta_i$ and $\mathrm{cr}\{\lambda_2 \leq \tilde{c}_i\} < \beta_i$;

Step 2. Let $\lambda = (\lambda_1 + \lambda_2)/2$;

Step 3. If $\mathrm{cr}\{\lambda \leq \tilde{c}_i\} \geq \beta_i$, let $\lambda_1 = \lambda$; otherwise, let $\lambda_2 = \lambda$.

Step 4. If $|\lambda_1 - \lambda_2| > \delta$ (a given small positive number), go to Step 2; otherwise, let $\lambda^* = \lambda_1$, output λ^*. then, the constraint $\mathrm{cr}\{\sum_{j=1}^{p_i} c_{ij} q_{ij} \leq \tilde{c}_i\} \geq \beta_i$ is equivalent to $\sum_{j=1}^{p_i} c_{ij} q_{ij} \leq \lambda^*$ approximately.

4. Hybrid intelligent algorithm for uncertain bi-level programming model

Generally speaking, uncertain bi-level programming models are difficulty to solve by traditional methods. A good way is to design some hybrid intelligent algorithms for solving them. In this section, we integrate the simplex method, stochastic simulation, fuzzy simulation algorithm and GA to produce a hybrid intelligent algorithm to obtain an approximate optimal solution. In this hybrid algorithm, a genetic algorithm serves a role of seeking the best facility location strategy, the simplex method is to find out the optimal possibility (allocation) of each OD pair to choose the path. And stochastic simulation and fuzzy simulation algorithm are used to compute uncertain functions. This algorithm will reduce the computation greatly, which makes it possible to deal with problems of quite large size.

We describe the algorithm as the following procedure:

Step 1. Initialize *popsize* chromosomes $X_l = (x_1^l, x_2^l, \ldots, x_n^l)$, $l = 1, 2, \ldots, popsize$. x_k, $k = 1, 2, \ldots, n$ are 0 or 1. If $x_k = 1$, then a facility is located at node v_k; otherwise, $x_k = 0$. If the sum of nonzero elements in X_l is not larger than m, then it will be treated as a feasible solution.

Step 2. According to X_l, we can obtain u_{ij}^l of path P_i^j for each OD pair. Compute λ^* of constraint (3.11) by fuzzy simulations, respectively. The simplex method is used to solve the linear programming $(L1)$ for each OD pair (v_{si}, v_{ti}) $(i = 1, 2, \ldots, r)$.

Step 3. Put the q_{ij}^l $(i = 1, 2, \ldots, r; j = 1, 2, \ldots, p_i)$ of $(L1)$ into the model $(U1)$. Calculate the objective value $\bar{f}^l$ for all chromosomes X_l, $l = 1, 2, \ldots, popsize$ by stochastic simulations.

Step 4. Compute the fitness of all chromosomes X_l, $l = 1, 2, \ldots, popsize$. The rank-based evaluation function is defined as

$$\mathrm{Eval}(X_l) = \tau(1-\tau)^{l-1}, \quad l = 1, 2, \ldots, popsize, \tag{4.1}$$

where the chromosomes $X_1, X_2, \ldots, X_{popsize}$ are assumed to have been rearranged from good to bad according to their objective values $\bar{f}^l$ and $\tau \in (0, 1)$ is a parameter in the genetic system.

Step 5. Select the chromosomes for a new population. The selection process is based on spinning the roulette wheel characterized by the fitness of all chromosomes for *popsize* times, and each time we select a single chromosome. Thus we obtain *popsize* copies of chromosomes, denoted also by X_l, $l = 1, 2, \ldots, popsize$.

Step 6. The crossover operator is applied on a selected pair of parents producing two offspring. In order to determine the parents for crossover operation,we repeat the following process from $l = 1$ to *popsize*: generating a random real number θ from the interval $[0, 1]$, the chromosome X_l will be selected as a parent provided that $\theta < \rho_c$, where the parameter ρ_c is the probability of crossover. Then we group the selected parents $X_1', X_2', X_3', \ldots$ to the pairs (X_1', X_2'), (X_3', X_4'), …. Without loss of generality, let us illustrate the crossover operator on each pair. The standard crossover usually randomly chooses crossover points and simply exchanges the segments of the parents' genetic codes. But this approach may produce incorrect offspring. The number of ones in an offspring may become different from m, although its parents had exactly m in their

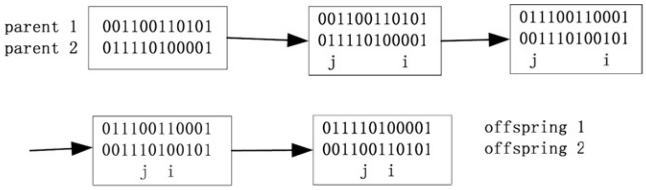

Fig. 1. Crossover operator.

Table 1
Computational results for static bi-level FIP.

n	*m*	*HBG*		*GA*			
		Solution values	Time	Popsize	Iteration	Solution values	Time
30	2	48	0.094	10	20	48	5.53
50	3	160	0.1560	10	20	160	10.5780
100	5	230	0.2810	20	20	230	34.2810
100	5	268	0.3280	20	50	255	90.3590
150	6	459	0.5160	20	100	452	100.7190
200	8	469	3.4530	20	100	469	393.50
200	10	507	4.219	20	100	507	425.76

genetic codes. To overcome this problem, the operator simultaneously traces the genetic codes of the parents from right to left searching the position i on which the first parent has 1 and second 0. The individuals exchange genes on the found position (identified as crossover point), and a similar process is performed starting from the left side of the genetic codes. The operator is searching the position j where the first parents has 0 and the other 1. Genes are exchanged on the jth position and the number of located facilities remains unchanged. This process is repeated until $j \leq i$. The parents are replaced with their offspring. (See Fig. 1.)

Step 7. The mutation operator is applied to update the chromosomes. Similarly, we repeat the following steps from $l = 1$ to *popsize*: generating a random real number θ from the interval $[0, 1]$, the chromosome X_l will be selected as parent provided that $\theta < \rho_m$, where the parameter ρ_m is the probability of mutation.The mutation is performed by changing a randomly selected gene in the genetic code $(0 - 1, 1 - 0)$. Finally, *popsize* new chromosomes may be generated, and we still denote them by X_l, $l = 1, 2, \ldots, popsize$.

Step 8. Repeat the second to seventh step for a given number of iterations.

Step 9. Report the best chromosome X^* as the optimal locations.

5. Computational results

In order to illustrate the effectiveness of the hybrid intelligent algorithm, this section will generate examples randomly to show the application of the model and algorithm. We first describe the test problem generation.

5.1. Random instance generation

To obtain random instances, for a given value of n, we first generate n points in $[0, 100]^2$ according to a continuous uniform distribution, and we then construct a full undirected graph G over these points, the length of an edge being the Euclidean distance between its two end points. Using a Prim algorithm, we then determine a shortest spanning tree G' over graph G. Edges of $G \setminus G'$ are added to the $n - 1$ edges of G' in decreasing order of length until a given threshold equal to $2n$ is reached. So the network G is constructed.

Then the number of OD pairs is $n/10$, origins and destinations are randomly selected from the n points. The flow of each OD pair is generated by discrete uniform distribution with in interval $[0, 50]$. We assume the customers only choose efficient paths to pass. Firstly, we give the definition of efficient paths. For a link (i, j), if node i is nearer to original node v_{si} of OD (v_{si}, v_{ti}) than node j and node j is farther to destination node v_{ti} than node i, link (i, j) is an efficient link for OD pair (v_{si}, v_{ti}). A path which is composed of efficient links of (v_{si}, v_{ti}) is called an efficient path of OD pair (v_{si}, v_{ti}). According to this definition, we can obtained efficient paths of every OD pair with Dail algorithm [27,28].

5.2. Results for static bi-level FIP

For random instances we compare the solution values obtained by the heuristic based on a greedy search and genetic algorithm with certain parameters (that means GA without stochastic and fuzzy simulations). Our computational results are summarized in Table 1. The flow of each OD pairs is a static value generated by discrete uniform distribution with in

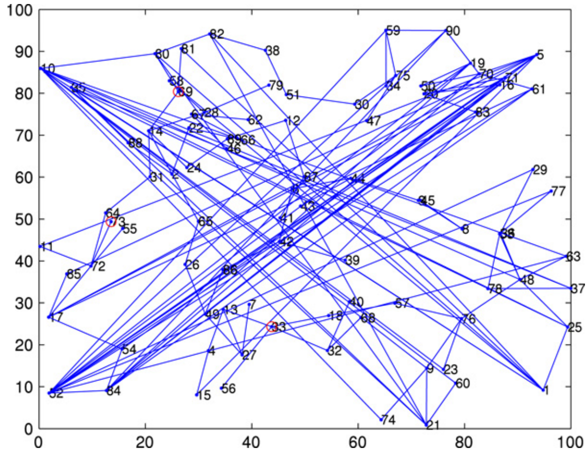

Fig. 2. A network with 90 nodes, the nodes to locate 3 facilities shown with red circle.

interval [0, 50]. Let L_i be the length of the shortest path of OD pair (v_{si}, v_{ti}). In the examples generated randomly, we assume the satisfied trip cost threshold c_i of OD pair (v_{si}, v_{ti}) be $1.4L_i$. The column headings are as follows.

n: number of vertices in the complete graph used to generate the instances;
m: number of facilities to be located;
HGS: heuristic based on greedy search;
GA: genetic algorithm;
time: the CPU time in seconds;
popsize: the size of chromosomes in GA;
iteration: iteration times in GA.

Results presented in Table 1 indicate that the heuristic based on a greedy search is an efficient algorithm to produce very good results. But for an uncertain bi-level programming model, the flow of every OD pairs is stochastic variable and the trip cost threshold level is fuzzy variables. We cannot use a heuristic based on a greedy search to solve it. So we integrate the simplex method, genetic algorithm, stochastic simulation and fuzzy simulation to develop a powerful hybrid intelligent algorithm.

5.3. Results for uncertain bi-level FIP

A network with 90 nodes generated randomly is shown in Fig. 2. Three facilities are located in this network to service the "pass-by" flow. The OD pairs generated randomly and the efficient paths of each OD pair obtained by Dial algorithm are shown in Fig. 3. fx_i is generated by discrete uniform distribution with in interval [0, 10] for OD pair (v_{si}, v_{ti}). ω is a stochastic variable which must obey the normal distribution $N(10, 2)$. So the flow of each OD pair $f_i = \omega * fx_i$. The satisfying trip cost threshold level of each OD pair (v_{si}, v_{ti}) is trapezoidal fuzzy number $(L_i, 1.2L_i, 1.4L_i, 1.6L_i)$. Let $\xi = (a, b, c, d)$ be a trapezoidal fuzzy number. Referring to Liu [29,30],we have

$$\mathrm{cr}\{\lambda \le \xi\} = \begin{cases} 1, & \lambda \le a, \\ \dfrac{2b-a-\lambda}{2(b-a)}, & a \le \lambda \le b, \\ \dfrac{1}{2}, & b \le \lambda \le c, \\ \dfrac{\lambda - d}{2(c-d)}, & c \le \lambda \le d, \\ 0, & \lambda \ge d. \end{cases} \tag{5.1}$$

In a stochastic simulation, denote simulation times $S = 3000$ and confidence level $\alpha = 0.7$. In a fuzzy simulation, let the fuzzy desired credibility of satisfying the trip cost threshold level $\beta_i = 0.8$ and the gap of termination $\delta = 0.01$. For a GA process, the parameter of fitness $\tau = 0.1$, the probability of crossover ρ_c and the probability of mutation ρ_m are 0.2.

According to computational results of a hybrid intelligent algorithm with these parameters, we obtain the nodes to locate three facilities (red circles in Fig. 2). The best solution values of the iteration are shown in Fig. 4.

We also test many different sized examples generated randomly with this hybrid intelligent algorithm coded in matlab 7.0. These tests show this algorithm is an efficient algorithm to solve this uncertain bi-level FIP. In Table 2, we compare solutions for this problem in the same network with 20 nodes when different parameters are taken with the same

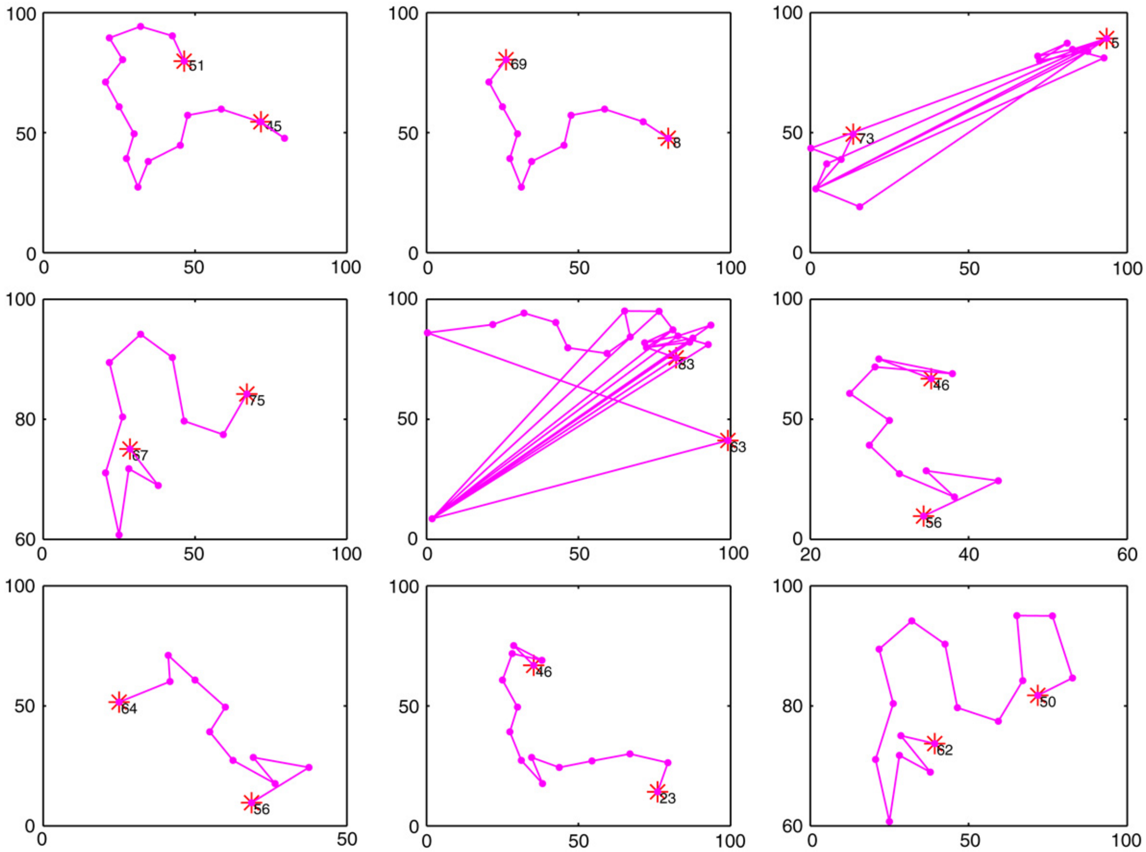

Fig. 3. OD pairs and efficient paths.

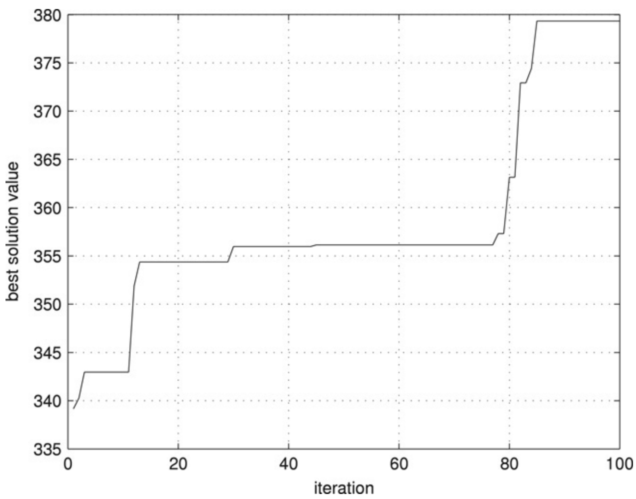

Fig. 4. Best solution values of iteration.

generations as a stopping rule. It appears that all the results differ little from each other. In order to account for it, we present a parameter *error* called the percent error, i.e. $(actualvalue - optimal\ value)/optimal\ value \times 100\%$, where optimal value is the largest one of all the ten results. It follows from Table 2 that the percent error does not exceed 0.28% when different parameters are selected. It implies that the hybrid intelligent algorithm is robust to the parameter settings and effective for solving this problem.

From model (U2) We find that the optimal objective is also dependent on the value of parameter α, thus it is meaningful to investigate the sensitivity of approximate optimal objectives with respect to α. We choose five values of α for a same

Table 2
Comparison solutions of an example ($\alpha = 0.2$).

Popsize	ρ_c	ρ_m	τ	Gen	Error (%)
10	0.3	0.1	0.08	100	0.23
10	0.3	0.2	0.05	100	0.04
10	0.1	0.2	0.05	100	0.13
10	0.1	0.3	0.10	100	0.28
10	0.2	0.2	0.10	100	0.00
20	0.1	0.2	0.05	200	0.12
20	0.2	0.2	0.10	200	0.18
20	0.1	0.3	0.10	200	0.01
20	0.3	0.1	0.08	200	0.19
20	0.3	0.2	0.05	200	0.27

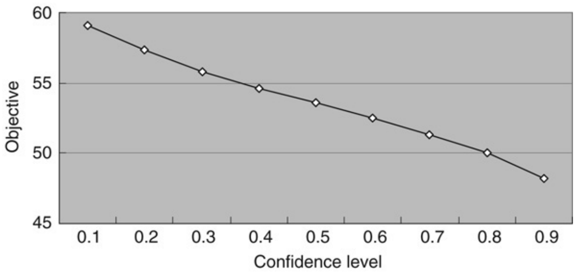

Fig. 5. The sensitivity of optimal objective with different α.

problem, that is, $\alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$. The approximate optimal objectives are listed in Fig. 5. It is easy to see that the objective will decrease with an increasing value of α.

6. Conclusion

In this paper, the facility location problem with customers' path choice is presented. We need to located m facilities in order to maximize the intercepted customers, while accounting for "flow-by" customers' preferences and their travel cost threshold. This problem is defined as a bi-level programming static model. Consequently, a heuristic based on a greedy search is designed to solve it. Due to the co-existence of randomness and fuzziness in the real world, we propose a chance constrained bi-level model with stochastic flow and fuzzy trip cost threshold level. For solving this bi-level uncertain model more efficiently, the simplex method, genetic algorithm, stochastic simulation and fuzzy simulation are integrated to design a hybrid intelligent algorithm. Some numerical examples are generated randomly to illustrate the performance and the effectiveness of the proposed algorithms. This modeling work focuses on game theory between the network planner and users. Future research should expand the model frame work to capacitated facilities, set-up cost of facilities and expected losses of users.

Acknowledgements

The authors thank the support of the National Science Foundation of China under Grants 70601011, the Program for New Century Excellent Talents in University of China under the Grant No. NCET-06-0653 and the Postdoctoral Foundation of HUST.

References

[1] M.S. Daskin, Network and Discrete Location: Models Algorithms and Applications, Wiley, New York, 1995.
[2] S.L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph, Operations Research 12 (1964) 450–459.
[3] C. Toregas, R. Swaim, The location of emergency service facilities, Operation Research 19 (1971) 1363–1373.
[4] R.L. Church, C. ReVelle, The maximal covering location problem, Papers of Regional Science Association 32 (1974) 101–118.
[5] M.J. Hodgson, A flow capturing location–allocation model, Geographical Analysis 22 (1990) 270–279.
[6] O. Berman, R.C. Larson, N. Fouska, Optimal location of discretionary service facilities, Transportation Science 26 (1992) 201–211.
[7] O. Berman, J. Hodgson, D. Krass, Facility Location: A Survey of Applications and Methods, 3rd ed., Springer-Verlag, Berlin Heidelberg, New York, 1995.
[8] M.J. Hodgson, K.E. Rosing, A.L.G. Stornier, Applying the flow capturing location–allocation model to an authentic network: Edmonton, Canada, European Journal of Operational Research 90 (1996) 427–443.
[9] O. Berman, D. Bertsimas, R.C. Larson, Locating discretionary service facilities, II: Maximizing market size, minimizing inconvenience, Operations Research 43 (1995) 623–632.
[10] M. Wen, K. Iwamura, Facility location–allocation problem in random fuzzy environment: Using (α, β)-cost minimization model under the Hurewicz criterion, Computers & Mathematics with Applications 55 (2008) 704–713.
[11] O. Berman, J. Wang, Z. Drezner, G. Wesolowsky, A probability minimax location problem on the plane, Annals of Operations Research 122 (2003) 59–70.

[12] S. Shiode, Z. Drezner, A competitive facility location problem on a tree network with stochastic weights, European Journal of Operation Research 149 (2003) 47–52.
[13] J. Zhou, B. Liu, New stochastic models for capacitated location–allocation problem, Computers and Industrial Engineering 45 (2003) 111–125.
[14] O. Berman, D. Krass, C.W. Xu, Locating flow-intercepting facilities: New approaches and results, Annals of Operations research 60 (1995) 121–143.
[15] Q. Wang, R. Batta, Algorithms for a facility location problem with stochastic customer demand and immobile servers, Annals of Operations Research 111 (2000) 17–34.
[16] T.H. Wu, J.N. Lin, Solving the competitive discretionary service facility location problem, European journal of Operational Research 144 (2003) 366–378.
[17] J. Yang, Facility location problems about flow interception in network, Ph.D. Thesis, Huazhong University of Science & Technology, China, 2005 (in Chinese).
[18] M. Simaan, J.B. Cruz, On the Stackelberg strategy in nonzero-sum games, Journal of Optimization Theory and Applications 11 (1973) 533–555.
[19] B. Hansen, B. Jaumard, G. Savard, New branch and bound rules for linear bilevel programming, SIAM Journal on Scientific and Statistical Computing 13 (1992) 1194–1217.
[20] B. Colson, P. Marcotte, G. Savard, Bilevel programming: A survey, 4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies 3 (2005) 87–107.
[21] J.T. Bard, J.T. Moore, An algorithm for the discrete bilevel programming problem, Naval Research Logistics 39 (1992) 419–435.
[22] J.T. Bard, J.T. Moore, The mixed integer linear bilevel programming problem, Operations Research 38 (1990) 911–921.
[23] B.D. Liu, Stackelberg–Nash equilibrium for multilevel programming with multiple followers using genetic algorithms, Computers & Mathematics with Applications 36 (1998) 79–88.
[24] M. Gendreau, P. Marcotte, G. Savard, A hybrid Tabu–Ascent algorithm for the linear bilevel programming problem, Journal of Global Optimization 9 (1996) 1–14.
[25] X.Y. Li, P. Tian, X.P. Min, A hierarchical particle swarm optimization for solving bilevel programming problems, in: ICAISC 2006, in: LNAI, vol. 4029, 2006, pp. 1169–1178.
[26] U. Takeshi, K. Hideki, Single and multi-objective defensive location problems on a network, European Journal of Operational Research 188 (2008) 76–84.
[27] R.B. Dial, Algorithm of shortest path forest with topological ordering, Communications of the ACM 12 (1969) 632–633.
[28] M.G.H. Bell, Alternatives to Dial's logit assignment algorithm, Transportation Research B 29 (1995) 287–295.
[29] B. Liu, Theory and Practice of Uncertain Programming, Physica-Verlag, New York, 2002.
[30] B. Liu, Uncertainty Theory, 2nd ed., Springer-Verlag, Berlin, 2007.